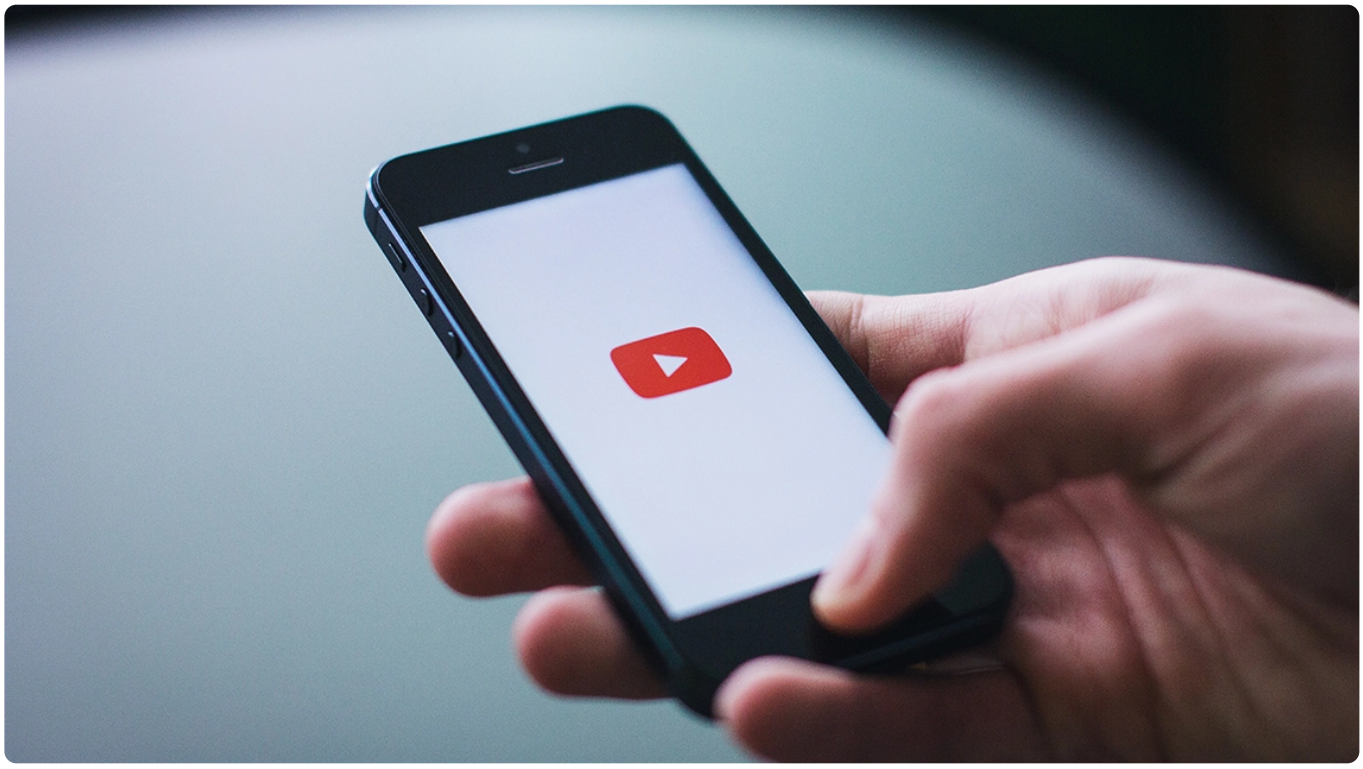
14 | MP4 & FLV：不要再说AVI了

2021-12-22 李江

《攻克视频技术》 课程介绍 >



讲述：李江
时长 11:53 大小 10.90M

你好，我是李江。

前面我们花了很长的时间学习了视频编码和视频传输弱网对抗的知识点。从今天开始我们来学习几个视频封装和播放的知识点。我们先来学习一下什么是音视频封装。之后我们再学习如何做音视频同步。

其实相比视频编码和传输，音视频封装应该是非常简单的知识点了。而且我们前面还学习过 RTP 打包，RTP 打包音视频数据其实一定程度上也可以算是一种封装。我们今天再介绍两种常用的封装，一种是 FLV，一种是 MP4，相信你对这两种文件一点儿也不陌生。

音视频封装其实就是将一帧帧视频和音频数据按照对应封装的标准有组织地存放在一个文件里面，并且再存放一些额外的基础信息，比如说分辨率、采样率等信息。那到底怎么组

织这些基础信息还有音视频数据呢？我们接下来先看看 FLV 是怎么做的。

FLV

FLV 是一种非常常见的音视频封装，尤其是在流媒体场景中经常用到。FLV 封装也是比较简单的封装格式，它是由一个个 Tag 组成的。Tag 又分为视频 Tag、音频 Tag 和 Script Tag，分别用来存放视频数据、音频数据和 MetaData 数据。

下图就是 FLV 的总体结构图：

其总体格式图如下：

	字段	占用大小	含义	
FLV Header	Signature	3字节	标识符，值为 “FLV”	
	Version	1字节	版本号	
	Flags	1字节	第6位表示音频是不是存在，第8位表示视频是不是存在，其余为0	
	Header Size	4字节	头部大小，一般为9字节	
FLV Body	Previous Tag Size	4字节	前一个Tag的大小	
	Tag 1		Tag Header	Type：1字节，表示Tag的类型，音频是0x08，视频是0x09，Script是0x12
				Data Size：4字节，表示Tag Data的大小
				Timestamp：3字节，时间戳
				TimestampEx：1字节，扩展时间戳，Timestamp不够的时间使用
				StreamId：1字节，值为0
			Tag Data	具体的数据
	Previous Tag Size	4字节	前一个Tag的大小	
	.....			
	Previous Tag Size	4字节	前一个Tag的大小	

FLV Header

其中，FLV Header 占用 9 个字节。前 3 个字节是文件的标识，固定是 FLV。之后的 1 个字节表示版本。在之后的 1 个字节中的第 6 位表示是否存在音频数据，第 8 位表示是否存在视频数据，其他位都为 0。最后的 4 个字节表示从文件开头到 FLV Body 开始的长度，一般就是等于 9。

FLV Body

在 FLV Header 之后就是 FLV Body 了，这就是存放主要数据的地方，放置着一个个 Tag。在每一个 Tag 前面都有一个 4 字节的 Previous Tag Size，表示前一个 Tag 的大小，方便往回倒。再之后就是具体的 Tag 了。Tag 又是由 Tag Header 和 Tag Data 组成，其中 Tag Header 占用 11 个字节，格式如上图。

其中最重要的是时间戳，因为播放的速度还有音视频同步都需要依赖这个时间戳的值。时间戳占用 3~4 字节，如果 3 字节不够的话，则需要使用 1 字节的扩展时间戳作为时间戳的高 8 位。还需要注意的一个点就是，时间戳的单位是 ms。RTP 的时间戳单位是

1/90000 秒，MP4 的时间戳是可以自定义的。这个时间戳的单位也是至关重要的，不要弄错了。

接下来就是 Tag Data 数据了。Tag Data 有 Script、音频和视频。首先来看一下 Script Tag 的 Data。这个 Tag 存放的是 MetaData 数据，主要包括宽、高、时长、采样率等基础信息。

Script Data 使用 2 个 AMF 包来存放信息。第一个 AMF 包是 onMetaData 包。第 1 个字节表示的是 AMF 包的类型，一般是字符串类型，值是 0x02，之后是 2 字节的长度，一般长度总是 10，值是 0x000A。之后就是 10 字节长度字符串了，值是 onMetaData。

第二个 AMF 包的第一个字节是数组类型，值是 0x08，紧接着 4 个字节为数组元素的个数。后面即为各数组元素的封装，数组元素为元素名称和值组成的对。常见的数组元素如下表所示：

元素类型	元素的意义
duration	时长
file size	文件大小
width	视频的宽
height	视频的高
video data rate	视频码率
frame rate	视频帧率
video codec id	视频编码类型
audio sample rate	音频采样率
audio sample size	音频的位宽
audio data rate	音频编码码率
stereo	是不是立体声
audio codec id	音频编码类型

音频 Tag Data 的第一个字节表示音频的编码方式、采样率和位宽等信息，如下图所示。之后就是音频数据了。

字段	占用大小	说明
音频编码类型	4bits	2表示mp3 10表示aac
采样率	2bits	0表示5.5kHz 1表示11kHz 2表示22kHz 3表示44kHz
位宽	1bit	0表示8位 1表示16位
声道数	1bit	0表示单声道 1表示双声道

视频 Tag 的第 1 个字节包含了这个 Tag 的视频帧类型和视频编码方式，格式如下图：

字段	占用大小	说明
视频帧类型	4bits	1表示关键帧 2表示非关键帧
视频编码类型	4bits	7表示H264 AVC

对于 H264 数据，紧接着会有 4 字节的 AVC Packet Type 格式，如下图所示：

字段	占用大小	说明
AVC 包类型	1字节	0表示序列头（包含SPS PPS） 1表示NALU
CTS	3字节	如果包类型是0则为0 如果包类型是1则表示cts cts = pts - dts

其中最重要的就是 CTS。这个是什么意思呢？这是因为 H264 有 B 帧这种类型，涉及到显示时间戳 PTS 和解码时间戳 DTS。前面 Tag Header 里的时间戳就是 DTS，PTS 等于 DTS + CTS，这个需要注意一下。接下来就是存放具体的视频数据。

如果 AVC 包类型是 0，则数据格式如下图所示：

字段	占用大小	说明
版本号	1字节	版本号为1
AVC Profile Indication	1字节	H264 Profile
Profile Compatibility	1字节	
AVC Level Indication	1字节	H264 Level
Length Size Minus One	1字节	
SPS的个数	1字节	
SPS的大小	2字节	
SPS	具体大小 前面计算得到	
PPS的个数	1字节	
PPS的大小	2字节	
PPS	具体大小 前面计算得到	

如果 AVC 包类型为 1，则数据格式如下图所示：

字段	占用大小
NALU的长度	4字节
NALU数据	具体大小 前面计算得到

这就是 FLV 封装。

MP4

了解了 FLV 封装之后，我们再来看一下 MP4 封装。MP4 封装相比 FLV 更常见，但是也更复杂一些。其实它们的基本的思想还是一样的，就是用一个规定的格式组织存放音视频数据和一些基础信息。跟 FLV 由一个个 Tag 组成有点类似，MP4 由一个个 box 组成，每一个 box 存放了不同的数据，而且 box 里面还可以嵌套着 box。

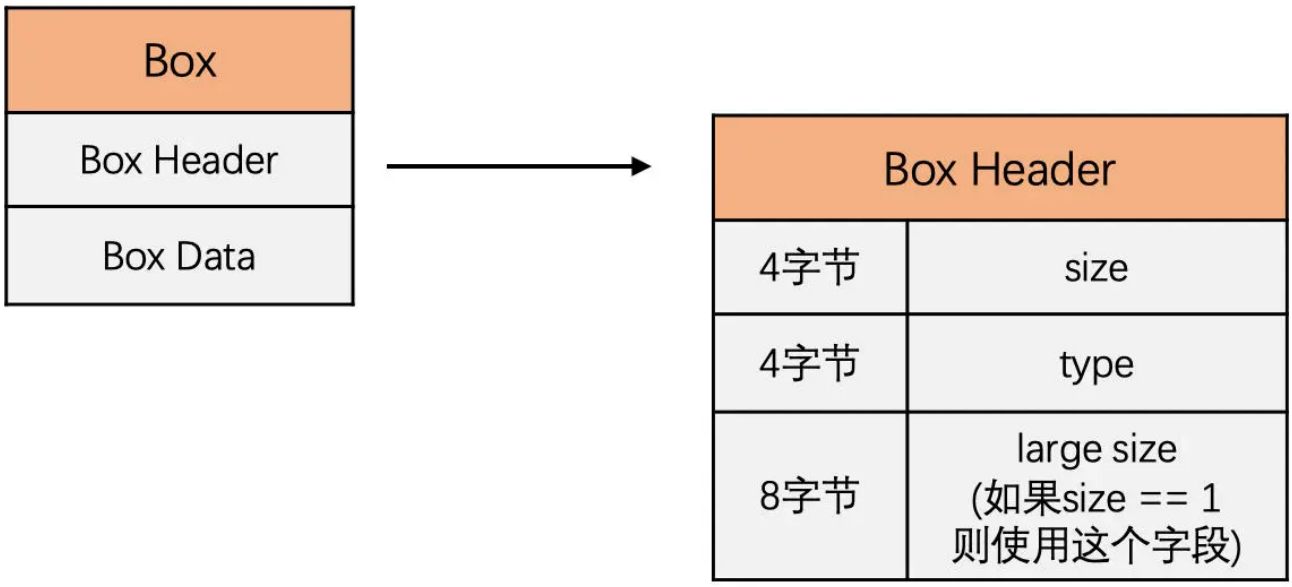
MP4 最外层的 box 主要有三个，分别是 File Type box（ftyp box）、Movie box（moov box）和 Media Data box（mdat box）。其中最重要、最复杂的就是 moov box 了，它里面存放了音视频的基本信息和每一个音视频数据的具体位置。

还有一点需要说明的就是：在 MP4 文件中，视频的一帧和音频的一段编码数据称为一个 sample。连续的几个 sample 称之为 chunk，而视频的所有 sample 称为一个视频 track，同样音频的所有 sample 也称为一个音频 track。

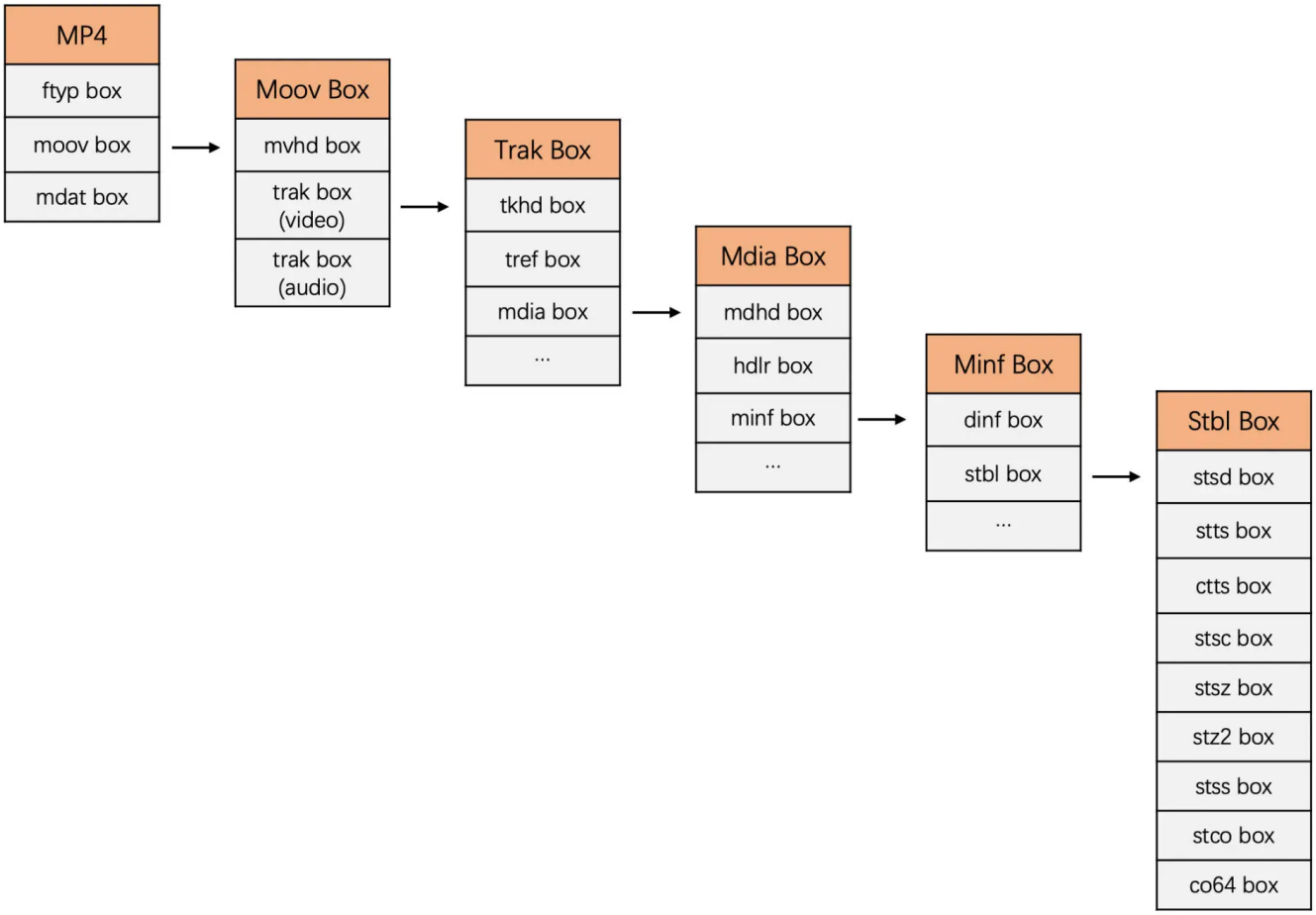
因此一般 MP4 文件是由音频 track 和视频 track 组成，而 track 由 sample 组成，其中若干个 sample 组成一个 chunk。

好了，下面我们就来看看比较重要的 box 吧。因为，MP4 的 box 特别多，我们不会一个个都介绍，我们只介绍一下比较重要的 box。

每一个 box 都是由 Box Header 和 Box Data 组成。Box Header 的结构如下图所示：



根据 Box Header 中的 type 我们将 box 分为不同类型的 box，每一种不同的 box 对应的 Box Data 都是不一样。Box Data 里面又可以嵌套 box。MP4 的总体 box 分布图如下图所示：



首先，ftyp box 放在 MP4 文件的开始，用来表示文件类型，该 box 的 Box Data 包含了 4 字节的主版本（major brand）、4 字节的版本号（minor version）和若干个 4 字节数

组组成的兼容版本（compatible_brands）。

mdat box 是 MP4 的音视频数据存放的地方。mdat box 基本由头部和数据两部分组成，box type 是 “mdat” 的 ASCII 码值。对于 H264 来说，是一个个 NALU，码流格式使用的是第 05 讲里面的 MP4 格式。这里的 NLAU 不再包含 SPS 和 PPS，这些数据已经放到 moov box 里面了，此处 NALU 类型是图像数据或者 SEI 数据。

另一个 box 就是最重要的 moov box，用来存放 Metadata 信息。这个 box 可以放在 ftyp 的后面也可以放置在文件的最后面。moov box 里面会一层层嵌套很多层 box。总体嵌套逻辑就是 movie 里面是 track，track 里面是 sample，多个 sample 又组成了一个个 chunk。

moov box 首先有一个 mvhd box（movie header box）主要存放文件的基本信息，比如说 MP4 文件的创建时间、时间单位、总时长等信息。

moov box 中的另外一个重要的 box 就是 trak box，这个 box 音频和视频各有一个。具体是音频 trak 还是视频 trak，会在 trak box 中的 mdia box 里面的 hdlr box 中表示出来。

trak box 内部有一个 tkhd box（track header box）主要是表示 track 的一些基本信息，比如说视频的宽高信息和音频的音量信息等。

trak box 还有一个 mdia box，它是媒体信息 box。它包含了 3 个子 box，一个是 mdhd box，一个是刚才提到的 hdlr box，一个是最重要的 minf box，这个 box 里面包含了 sample 的很多信息，这些信息是找到并正确使用音频和视频数据的关键。

mdhd box 里面最重要的一个值就是时间单位 time scale，这个时间单位是 sample 的时间戳的单位，控制播放速度和音视频同步都需要使用到这个值。

hdlr box 主要包含了 track 的类型信息，表明是音频还是视频 track。

minf box 里面包含了一个 stbl box（sample table box），里面存放着可以计算得到每一个 chunk 的偏移地址、每一个 sample 在文件中的地址信息和大小、每一个 sample 的时间戳和每一个视频 IDR 帧的地址信息。下面我们来详细介绍一下这些 box。

其中，stts box 中放置的是每一个 sample 的时长，这个值是 DTS。

字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	stts
版本	1字节	Box的版本
标志	3字节	Flags为0
条目数	4字节	Time to sample的数目
Time to sample 0		
Sample count	4字节	相同duration的sample数目
Sample Duration	4字节	Sample的时长
...		
Time to sample n		
Sample count	4字节	相同duration的sample数目
Sample Duration	4字节	Sample的时长

ctts box 放置着 CTS，也就是每一个 sample 的 PTS 和 DTS 的差值。

字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	ctts
版本	1字节	Box的版本
标志	3字节	Flags为0
条目数	4字节	Sample delta的数目
Sample delta 0		
Sample count	4字节	拥有相同delta的sample个数
Sample delta	4字节	pts减去dts的差值
...		
Sample delta n		
Sample count	4字节	拥有相同delta的sample个数
Sample delta	4字节	pts减去dts的差值

stss box 中放置的是哪些 sample 是关键帧。

字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	stss
版本	1字节	Box的版本
标志	3字节	Flags为0
条目数	4字节	Sync sample的数目
Sync Sample 0		
Sample序号	4字节	关键帧的Sample序号
...		
Sync Sample n		
Sample序号	4字节	关键帧的Sample序号

stsc box 中放置的是 sample 到 chunk 的映射表，也就是哪些 sample 属于哪个 chunk。

字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	stsc
版本	1字节	Box的版本
标志	3字节	Flags为0
条目数	4字节	Sample to chunk的数目
Sample to chunk 0		
First chunk	4字节	第一个chunk的序号
Samples per chunk	4字节	当前chunk中的sample个数
Sample description id	4字节	Sample的描述id
...		
Sample to chunk n		
First chunk	4字节	第一个chunk的序号
Samples per chunk	4字节	当前chunk中的sample个数
Sample description id	4字节	Sample的描述id

stco box 或 co64 box 中放置着每个 chunk 在文件中的偏移地址。

字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	stco或者co64
版本	1字节	Box的版本
标志	3字节	Flags为0
条目数	4字节	Chunk offset的数目
Chunk offset 0		
Chunk偏移	4字节（stco） 8字节（co64）	这个chunk相对于文件开头的偏移
...		
Chunk offset n		
Chunk偏移	4字节（stco） 8字节（co64）	这个chunk相对于文件开头的偏移

stsz box 中放置着每一个 sample 的大小。

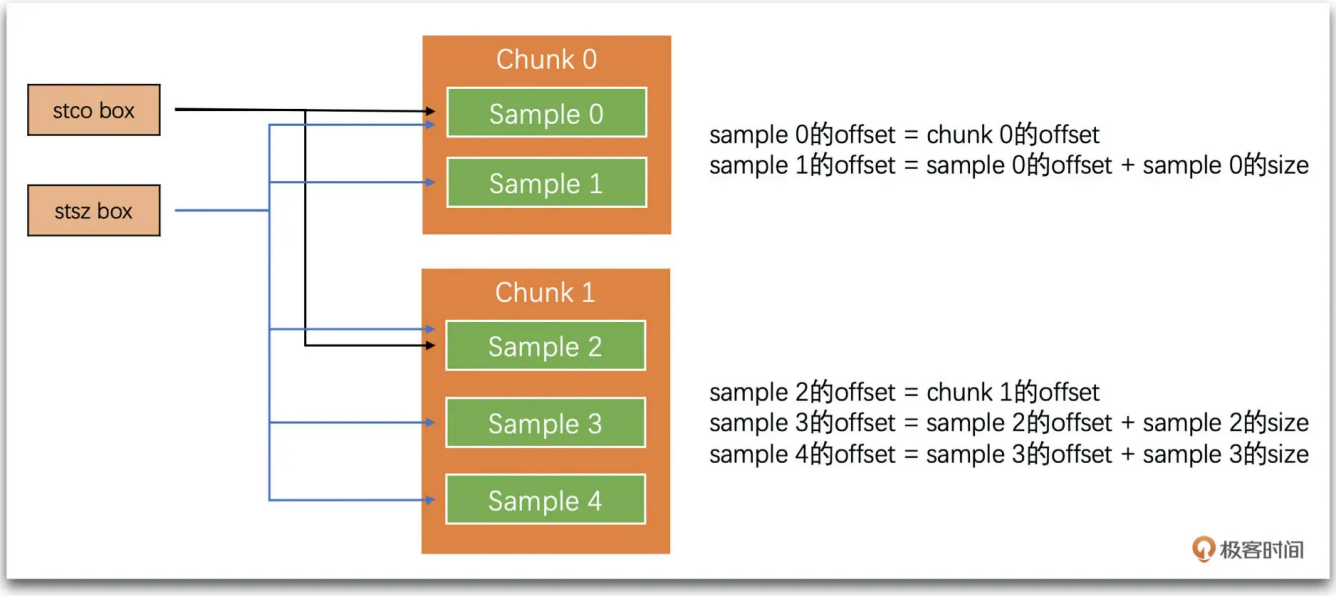
字段	占用大小	说明
大小	4字节	Box的字节数
类型	4字节	stsz
版本	1字节	Box的版本
标志	3字节	Flags为0
Sample size	4字节	如果所有的sample拥有相同的长度，那么大小就是这个值，如果不相同这个值就是0
条目数	4字节	Sample Size的数目
Sample size 0		
Sample size	4字节	第1个sample的大小
...		
Sample size n		
Sample size	4字节	第n个sample的大小

好了，跟 sample 相关的 box 就是这些。

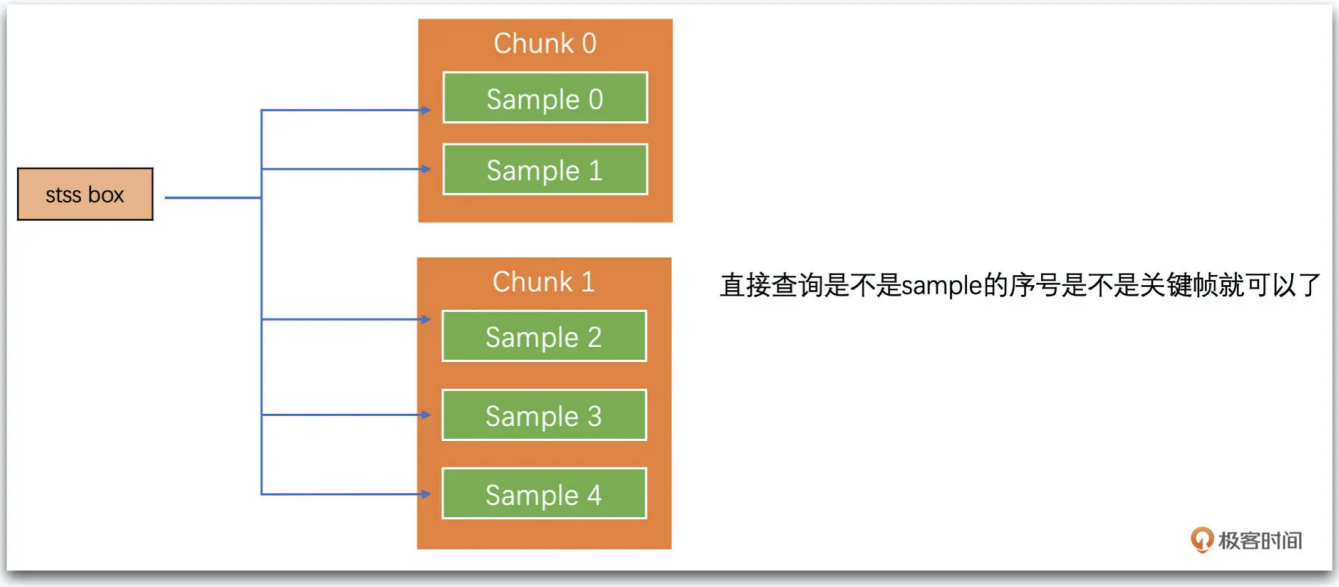
工程实践

接下来我们结合一个工程问题来实践一下。我们如何计算每一个 sample 在文件中的具体位置，判断它是不是关键帧，并计算它的具体时间。

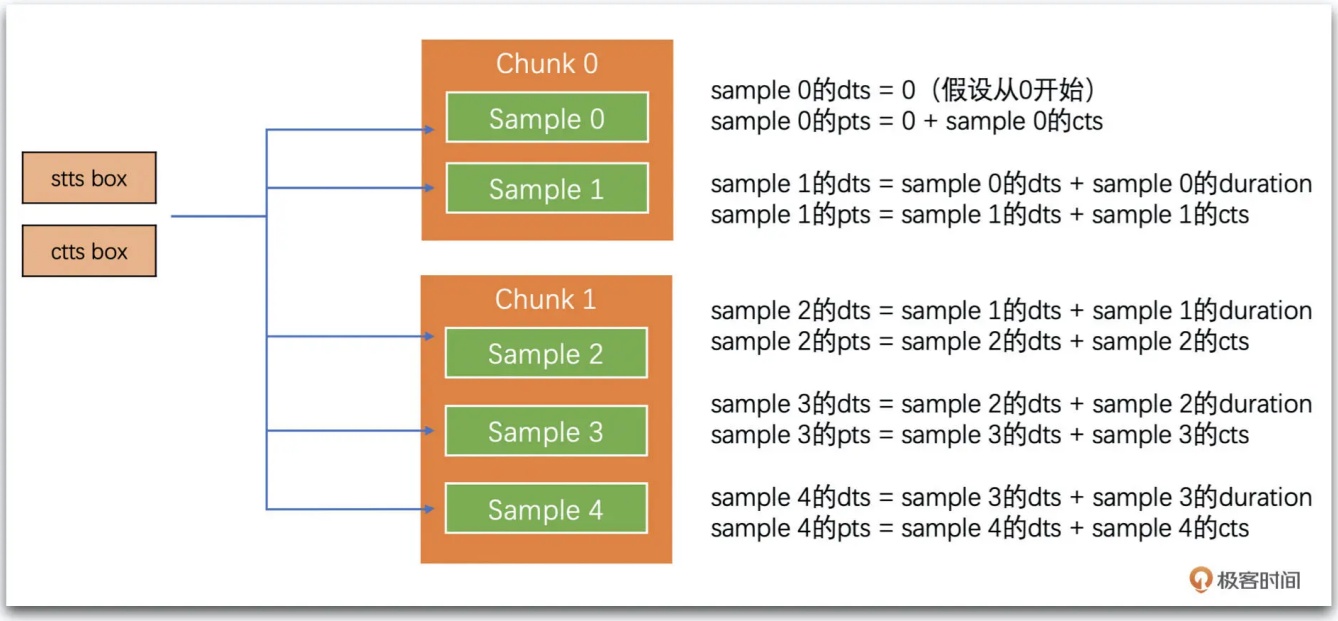
计算 sample 的具体位置需要使用 stco（或 co64）、stsc 和 stsz。我们首先通过 stsc 将每一个 sample 属于哪一个 chunk 计算出来。这样每一个 chunk 的第一个 sample 就知道是哪个了。然后我们通过 stco 和 co64 就可以知道对应序号的 chunk 的第一个 sample 在文件中的地址了。我们再通过 stsz 查询每个 sample 的大小，从 chunk 的第一个 sample 的地址将中间的 sample 的大小一个个地加上去就可以得到每一个 sample 的地址了。



而 sample 是不是关键帧，我们只需要通过 stss 对应每一个 sample 序号查询就可以得到。



计算 sample 的时间我们需要用到 stts 和 ctts。我们先通过 stts 得到每一个 sample 的时长，第 n 个 sample 的结束时间就是第 n-1 个 sample 的结束时间加上第 n 个 sample 的时长。但是需要注意一下，这个是 DTS，我们还需要通过 ctts box 得到每一个 sample 的 PTS 和 DTS 的差值。最后每一个 sample 的 PTS 就是等于 DTS 加上 CTS。



好了，以上就是 MP4 封装的主要内容。

总结

今天，我们主要介绍了一下两种音视频封装格式，分别是 FLV 和 MP4。这两种封装格式是我们工作和生活中经常需要用到的。

> FLV 在流媒体场景经常会用到，其实直播 RTMP 协议和 HTTP-FLV 协议里面也是用的 FLV 封装，所以还是很重要的。
>
> MP4 封装就是平时视频文件最常用的封装了，它主要由一个个 box 组成，其中最重要的就是跟 sample 有关的 box，你需要重点掌握。当然你也不需要背下来，了解主要思想即可，等真正用到的时候查询一下就可以了。

思考题

好了，今天的课程到这里就结束了。在课程的最后，我给你留一道思考题。请你想一想：为什么 FLV 相比 MP4 更适合流媒体？

你可以在留言区留下你的答案，或者你有任何问题都可以在留言区和我交流，我们下节课再见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 13 | SVC：如何实现视频编码可伸缩？

下一篇 加餐 | 基于纯浏览器的视频会议方案探究

精选留言 (3)

写留言

翻山越岭
2021-12-22

1)flv,mpeg和rmvb文件可以任意大小，不需要通过索引分包，任意位置可以直接解析分包
2)mp4，avi一定要依赖索引表才行，开始就要固定好位置，如果索引表在尾部，就没办法解析

2

陈诚
2021-12-24

请问一下：mp4文件中，sps，pps只是存储一组吗？后续来的sps，pps直接丢弃，还是sps，pps也可以存储多组？

晓龙
2021-12-22

MP4中stbl box中是视频编解码信息，mdat box中的音视频裸数据，只要拿到整个MP4文件，才能完整解码播放